MTHSC 4420 Advanced Mathematical Programming Test 1

Megan Bryant

November 12, 2013

Problem 1

Joe, Bob, and Bubba's rude radio morning show runs commercials in 60-second blocks. Somehow, they have managed to sell commercials for the 8-9am hour running 15, 16, 20, 25, 30, 35, 40, and 50 seconds. Formulate an integer program that can be used to determine the minimum number of 60-second blocks needed in order to run all the commercials and in which block each commercial will run. (Hint: Certainly no more than eight blocks are needed). Solve your program with AMPL.

IP Formulation

Let $f = (15, 16, 20, 25, 30, 35, 40, 50)$

Let $C = \{1, \cdots, 8\}$ be the set of commercials.

Let $B = \{1, \cdots, 8\}$ be the set of commercial blocks.

Let $y_j = \begin{cases} 1, & \text{if block } j \text{ is used} \\ 0, & \text{otherwise} \end{cases}, \forall j \in B.$

Let $x_{ij} = \begin{cases} 1, & \text{if commercial } i \text{ is assigned to block } j \\ 0, & \text{otherwise} \end{cases}, \forall i \in C, j \in B$

Minimize: $\sum\limits_{j \in B} y_j$

subject to:

$\sum\limits_{i \in C} f_i x_{ij} \leq 60 y_j$, for all $j \in B$

$\sum\limits_{j \in B} x_{ij} = 1$, for all $i \in C$

$x_{ij} \in \{0, 1\}$
$y_j \in \{0, 1\}$

AMPL Code

Mod file:

```
param commercials = 8;
param blocks = 8;
param max_time = 60;
param commercial_length{i in 1..commercials};

var y{j in 1..blocks} binary;
#1 if block j is used

var x{i in 1..commercials, j in 1..blocks} binary;
#1 if commercial i is assigned to block j

minimize Number_Blocks:
sum{j in 1..blocks}y[j];
#Minimize the number of blocks used

subject to Time_per_block {j in 1..blocks}:
sum{i in 1..commercials}commercial_length[i]*x[i,j]<=max_time*y[j];
#Ensures block time is less than max_time for each block

subject to Commercial_Aired_Once {i in 1..commercials}:
sum{j in 1..blocks} x[i,j] = 1;
#Ensures commercials are aired in only one block
```

Data file:

```
param commercial_length :=
1 15 2 16 3 20 4 25 5 30 6 35 7 40 8 50;
```

AMPL Solution

```
ampl: model problem1mod.mod;
ampl: data problem1dat.dat;
ampl: option solver gurobi;
ampl: solve;
Gurobi 5.5.0: optimal solution; objective 5
48 simplex iterations
ampl: display Number_Blocks;
Number_Blocks = 5
```

Problem 2

Consider the 0-1 knapsack problem:
max $\{\sum_{j=1}^{n} c_j x_j : \sum_{j=1}^{n} a_j x_j \leq b; x_j \in \{0,1\}, j = 1, \cdots, n\}$.

a.)*Assume that* $c_1/a_1 \geq c_2/a_2 \geq \cdots \geq c_n/a_n$ *and suppose* r *is such that* $\sum_{j=1}^{r-1} a_j \leq b$ *and* $\sum_{j=1}^{r} a_j > b$.
Show that the optimal solution of the LP relaxation is
$x_j = 1$ *for* $j = 1, \cdots, r-1, x_r = (b - \sum_{j=1}^{r-1} a_j)/a_r$ *and* $x_j = 0$ *for* $j = r+1, \cdots, n$.

By duality, we see that the primal is:
max $c^T x$
subject to: $a^T x \leq b$
$0 \leq x \leq 1$

Therefore, the dual is:
min $by + z$
subject to: $a_i y + z_i \geq c_i$
$0 \leq z$
$0 \leq y$

Let $y^* = c_n/a_n$ and $z_i \begin{cases} c_i - a_i y, & \text{if } i = 1, \cdots, r-1 \\ 0, & \text{if } i = r+1, \cdots, n \end{cases}$

Then the optimal solution of the LP relaxation is $x_j = 1$ for $j = 1$, $\cdots, r-1, x_r = (b - \sum_{j=1}^{r-1} a_j)/a_r$ and $x_j = 0$ for $j = r+1, \cdots, n$.

b.) *Use the branch and bound algorithm to solve:*
max $17x_1 + 10x_2 + 25x_3 + 17x_4$
s.t. $5x_1 + 3x_2 + 8x_3 + 7x_4 \leq 12$
$x_1, \cdots, x_4 \in \{0, 1\}$.
Draw the branch-and-bound tree. Label each node with the value of the relaxation solution. Label each branch with the constraint added on that branch. Branch first to the left at each node, imposing the round-up constraint first. Use a depth-first strategy, always branching from the most recently created live node. Note that at each node with a fractional solution, you can generate a feasible solution by rounding the fractional variable down. Use this fact along with fathoming by integrality to maintain the incumbent integer solution and lower bound. Record each integer solution that becomes the incumbent and the node where it was found.

First, we will solve the linear relaxation in AMPL.

Linear Relaxation AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
param constraint_one_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize Knapsack_Value:
sum{i in 1..num_vars}objective_values[i]*x[i];

subject to Constraint_one:
sum{i in 1..num_vars}constraint_one_values[i]*x[i] <= 12;

option relax_integrality 1;
```

AMPL Data:

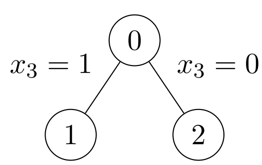
```
param objective_values := 1 17 2 10 3 25 4 17;

param constraint_one_values := 1 5 2 3 3 8 4 7;
```

Linear Relaxation AMPL Solution
AMPL Model:

```
ampl: model lpamplmod.mod;
ampl: data lpampldat.dat;
ampl: option solver gurobi;
ampl: solve;
Gurobi 5.5.0: optimal solution; objective 39.5
1 simplex iterations
ampl: display x;
x [*] :=
1  1
2  1
3  0.5
4  0
;
```

So, we will branch on x_3.

$\widetilde{v} = 39.5$, $\widetilde{x_3} = .5$, round = 35, $\hat{\text{v}} = 35$



First, we will solve the $x_3 = 1$ node.

Linear Relaxation AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
param constraint_one_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize Knapsack_Value:
sum{i in 1..num_vars}objective_values[i]*x[i];
```

```
subject to Constraint_one:
sum{i in 1..num_vars}constraint_one_values[i]*x[i] <= 12;

subject to Constraint_two:
x[3] = 1;

option relax_integrality 1;
```

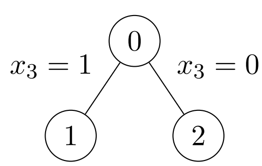
AMPL Data:

```
param objective_values := 1 17 2 10 3 25 4 17;

param constraint_one_values := 1 5 2 3 3 8 4 7;
```

Linear Relaxation AMPL Solution
AMPL Model:

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 38.6
x [*] :=
1  0.8
2  0
3  1
4  0
;
```

$\widetilde{v} = 39.5$, $\widetilde{x_3} = .5$, round $= 35$, $\hat{\text{v}} = 35$



$\widetilde{v^1} = 38.6$, $\widetilde{x_1^1} = .8$, round $= 35$

Since the rounding is equal to the $\hat{\text{v}} = 35$ it is the optimal with $\mathbf{x} = (0, 1, 1, 0)$ and we can terminate the branch with this node being the incumbent.

Next, we will solve the $x_3 = 0$ node.

Linear Relaxation AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
param constraint_one_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize Knapsack_Value:
sum{i in 1..num_vars}objective_values[i]*x[i];

subject to Constraint_one:
sum{i in 1..num_vars}constraint_one_values[i]*x[i] <= 12;

subject to Constraint_three:
x[3] = 0;

option relax_integrality 1;
```

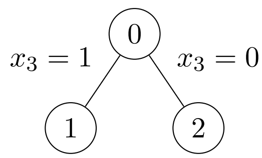
AMPL Data:

```
param objective_values := 1 17 2 10 3 25 4 17;

param constraint_one_values := 1 5 2 3 3 8 4 7;
```

Linear Relaxation AMPL Solution
AMPL Model:

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 36.71428571
1 simplex iterations
x [*] :=
1  1
2  1
3  0
4  0.571429
;
```

$\widetilde{v} = 39.5$, $\widetilde{x_3} = .5$, round $= 35$, $\hat{v} = 35$



$\widetilde{v^2} = 36.71428571$, $\widetilde{x_1^2} = .8$, $\widetilde{x_4^2} = .571429$ round $= 35$

Since the rounding is equal to the $\hat{v} = 35$, which is the incumbent. Therefore, we will add children nodes investigating the integrality of x_4.

Next, we will solve the $x_4 = 1$ node.

Linear Relaxation AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
param constraint_one_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize Knapsack_Value:
sum{i in 1..num_vars}objective_values[i]*x[i];

subject to Constraint_one:
sum{i in 1..num_vars}constraint_one_values[i]*x[i] <= 12;

subject to Constraint_three:
x[3] = 0;

subject to Constraint_four:
x[4] = 1;

option relax_integrality 1;
```

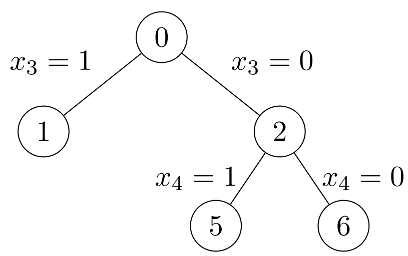
AMPL Data:

```
param objective_values := 1 17 2 10 3 25 4 17;

param constraint_one_values := 1 5 2 3 3 8 4 7;
```

Linear Relaxation AMPL Solution
AMPL Model:

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 34
x [*] :=
1  1
2  0
3  0
4  1
;
```

$\widetilde{v} = 39.5$, $\widetilde{x_3} = .5$, round $= 35$, $\hat{v} = 35$



$\widetilde{v^5} = 34$

Since $\widetilde{v^5} = 34$, we will fathom by bounds and terminate this branch.

Next, we will solve the $x_4 = 0$ node.

Linear Relaxation AMPL Code

AMPL Model code:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
param constraint_one_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize Knapsack_Value:
sum{i in 1..num_vars}objective_values[i]*x[i];

subject to Constraint_one:
sum{i in 1..num_vars}constraint_one_values[i]*x[i] <= 12;

subject to Constraint_three:
x[3] = 0;

subject to Constraint_four:
x[4] = 0;

option relax_integrality 1;
```

AMPL Data:

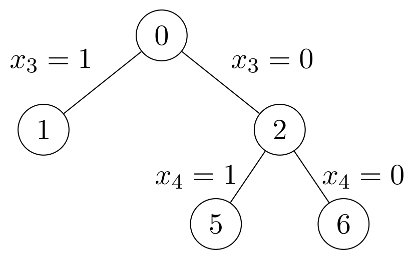
```
param objective_values := 1 17 2 10 3 25 4 17;

param constraint_one_values := 1 5 2 3 3 8 4 7;
```

Linear Relaxation AMPL Solution

AMPL Model:

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 27
x [*] :=
1  1
2  1
3  0
4  0
;
```

$\widetilde{v} = 39.5$, $\widetilde{x_3} = .5$, round $= 35$, $\hat{\text{v}} = 35$



$\widetilde{v^6} = 27$

Since $\widetilde{v^6} = 27 \leq 35$, we can fathom by bounds and terminate this branch.

Problem 3

Consider the knapsack contraint $\sum_{j=}^{n} a_j x_j \leq b$ *where*
$x_j \in \{0,1\}$ *for* $j = 1, \cdots, n$.
Recall that a cover is a set of indices $C \subseteq \{1, \cdots, n\}$ *such that* $\sum_{j \in C} a_j > b$.
The cover C *is minimal if* $\sum_{j \in C/\{i\}} a_j \leq b$ *for all* $i \in C$.
The cover inequality corresponding to a cover C *is* $\sum_{j \in C} x_j \leq |C| - 1$.

a.)*Show that the cover inequalities are valid for the set of solutions to the knapsack constraint.*

We know from definition 12.6 that a linear inequality is a valid inequality for a given discrete optimization model if it holds for all (integer) feasible solutions to the model.

Let there exist some x^* where $* \in B$, such that $\sum_{j \in C} x_j^* > |C| - 1$.

Then we know that $|B \cap C| = |C|$ which implies that $C \subseteq B$.

$\sum_{j=}^{n} a_j^* x_j = \sum_{j \in B} a_j \geq \sum_{j \in C} a_j > b$, by the given inequalities.

Therefore, $x^B \notin X$ and the cover inequalites are valid for the set of solutions to the knapsack constraints, since all feasible integer solutions hold to the model.

b.) *Given a minimal cover* C*, the extended cover* $E(C) = C \cup \{j : a_j \geq a_i \forall i \in C\}$.
The corresponding extended cover inequality is $\sum_{j \in E(C)} x_j \leq |C| - 1$.
For the problem in part 2b, write down all the extended cover constraints. Are any of these inequalities redundant? Solve the LP with constraints $0 \leq x_j \leq 1$ *for* $j = 1, \cdots, 4$ *and the nonredundant cover constraints. Compare the optimal value to the value of the root-node relaxation in part 2b.*

Original LP
max $17x_1 + 10x_2 + 25x_3 + 17x_4$
s.t. $5x_1 + 3x_2 + 8x_3 + 7x_4 \leq 12$
$x_1, \cdots, x_4 \in \{0, 1\}$.

Minimal Cover Constraints
$x_1 + x_2 + x_4 \leq 2$
$x_2 + x_3 \leq 1$

Therefore, $C = \{1, 2, 3, 4\}$

Extended Cover Constraints
$x_1 + x_2 + x_3 + x_4 \leq 2$
$x_1 + x_2 + x_3 + x_4 \leq 1$ REDUNDANT

Modified LP
max $17x_1 + 10x_2 + 25x_3 + 17x_4$
s.t. $5x_1 + 3x_2 + 8x_3 + 7x_4 \leq 12$
$x_1 + x_2 + x_3 + x_4 \leq 1$
$0 \leq x_1 \leq 1$
$0 \leq x_2 \leq 1$
$0 \leq x_3 \leq 1$
$0 \leq x_4 \leq 1$

AMPL Code

Model Ampl Code:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
param constraint_one_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize Knapsack_Value:
sum{i in 1..num_vars}objective_values[i]*x[i];

subject to Constraint_one:
sum{i in 1..num_vars}constraint_one_values[i]*x[i] <= 12;

subject to Constraint_two:
sum{i in 1..num_vars}x[i] <= 2;

option relax_integrality 0;
```

Model Ampl Code:

```
param objective_values := 1 17 2 10 3 25 4 17;

param constraint_one_values := 1 5 2 3 3 8 4 7;
```

AMPL Solution

Model Ampl Code:

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 35
4 simplex iterations
x [*] :=
1  0
2  1
3  1
4  0
;
```

The optimal objective for the method of extended covers is 35, which was the same optimal value we found using the branch and bound method.

Problem 4

$min\ 15x_{1.1} + 10x_{1.2} + 30x_{2.1} + 20x_{2.2}$
$s.t.\ x_{1.1} + x_{1.2} = 1$
$x_{2.1} + x_{2.2} = 1$
$30x_{1.1} + 50x_{2.1} \leq 80$
$30x_{1.2} + 50x_{2.2} \leq 60$
$x_{1.1}, x_{1.2}, x_{2.1}, x_{2.2} = 0\ or\ 1$

a.) *Use total enumeration to compute an optimal solution.*

Table 1: Total Enumeration

Case	Objective	Case	Objective
x = (0, 0, 0, 0)	Infeasible	**x** = (0, 1, 0, 1)	Infeasible
x = (1, 0, 0, 0)	Infeasible	**x** = (0, 0, 1, 1)	Infeasible
x = (0, 1, 0, 0)	Infeasible	**x** = (0, 1, 1, 0)	40
x = (0, 0, 1, 0)	Infeasible	**x** = (1, 1, 1, 0)	Infeasible
x = (0, 0, 0, 1)	Infeasible	**x** = (0, 1, 1, 1)	Infeasible
x = (1, 1, 0, 0)	Infeasible	**x** = (1, 0, 1, 1)	Infeasible
x = (1, 0, 1, 0)	45	**x** = (1, 1, 0, 1)	Infeasible
x = (1, 0, 0, 1)	35	**x** = (1, 1, 1, 1)	Infeasible

Therefore, by the total enumeration method, we can determine the optimal solution to be 45 when **x**=$(1, 0, 1, 0)$.

The following AMPL code was used to generate the solutions from the table:

AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize General_Assign:
sum{i in 1..num_vars}objective_values[i]*x[i];
```

```
subject to Constraint_One:
x[1] + x[2] = 1;

subject to Constraint_Two:
x[3] + x[4] = 1;

subject to Constraint_Three:
30*x[1] + 50*x[3] <= 80;

subject to Constraint_Four:
30*x[2] + 50*x[4] <= 60;

subject to Enumeration_One:
x[1] = 1;

subject to Enumeration_Two:
x[2] = 1;

subject to Enumeration_Three:
x[3] = 1;

subject to Enumeration_Four:
x[4] = 1;

option relax_integrality 0;
```

AMPL Data File:

```
param objective_values := 1 15 2 10 3 30 4 20;
```

AMPL Run File:

```
reset;
reset;
model problem4mod.mod;
data problem4dat.dat;
option solver gurobi;
solve;
```

b.) *Form a Lagrangian relaxation dualizing the third and fourth constraints with Lagrange multipliers v_1 and v_2.*

min $15x_{1.1} + 10x_{1.2} + 30x_{2.1} + 20x_{2.2} + v_1(1 - x_{1.1} - x_{1.2}) + v_2(1 - x_{2.1} + x_{2.2})$
s.t. $30x_{1.1} + 50x_{2.1} \leq 80$
$30x_{1.2} + 50x_{2.2} \leq 60$
$x_{1.1}, x_{1.2}, x_{2.1}, x_{2.2} = 0$ or 1

c.) *Explain how the dualization in part (b) leaves a relaxation that is easier to solve than the full ILP.*

The relaxation is now easier to solve than the full Integer Linear Program since the optimizations are now independent for the variables and the integrality constraints are relaxed.

d.) *Use total enumeration to solve the Lagrangian relaxation in part (b) with $v_1 = v_2 = 0$, and verify that the relaxation optimal value provides a lower bound on the true optimal value computed in part (a).*

min $15x_{1.1} + 10x_{1.2} + 30x_{2.1} + 20x_{2.2} + 0(1 - x_{1.1} - x_{1.2}) + 0(1 - x_{2.1} + x_{2.2})$
s.t. $30x_{1.1} + 50x_{2.1} \leq 80$
$30x_{1.2} + 50x_{2.2} \leq 60$
$x_{1.1}, x_{1.2}, x_{2.1}, x_{2.2} = 0$ or 1

AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize General_Assign:
sum{i in 1..num_vars}objective_values[i]*x[i] + 0*(1-x[1] - x[2]) + 0*(1-x[3] - x[4])

subject to Constraint_Three:
30*x[1] + 50*x[3] <= 80;

subject to Constraint_Four:
30*x[2] + 50*x[4] <= 60;

option relax_integrality 0;
```

AMPL Data File:

```
param objective_values := 1 15 2 10 3 30 4 20;
```

AMPL Run File:

```
reset;
reset;
model problem4mod.mod;
data problem4dat.dat;
option solver gurobi;
solve;
```

AMPL Solution

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 65
x [*] :=
1  1
2  0
3  1
4  1
;
```

The optimal of the lagrangian relaxation provides a lower bound of 65 on the true optimal of 45.

e.)*Repeat part (d) with* $v_1 = 10, v_2 = 12$*.*

min $15x_{1.1} + 10x_{1.2} + 30x_{2.1} + 20x_{2.2} + 10(1 - x_{1.1} - x_{1.2}) + 12(1 - x_{2.1} + x_{2.2})$
s.t. $30x_{1.1} + 50x_{2.1} \leq 80$
$30x_{1.2} + 50x_{2.2} \leq 60$
$x_{1.1}, x_{1.2}, x_{2.1}, x_{2.2} = 0$ or 1

AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};
```

```
var x{i in 1..num_vars} binary;

maximize General_Assign:
sum{i in 1..num_vars}objective_values[i]*x[i] + 10*(1-x[1] - x[2]) + 12*(1-x[3] - x[4

subject to Constraint_Three:
30*x[1] + 50*x[3] <= 80;

subject to Constraint_Four:
30*x[2] + 50*x[4] <= 60;

option relax_integrality 0;
```

AMPL Data File:

```
param objective_values := 1 15 2 10 3 30 4 20;
```

AMPL Run File:

```
reset;
reset;
model problem4mod.mod;
data problem4dat.dat;
option solver gurobi;
solve;
```

AMPL Solution

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 53
x [*] :=
1  1
2  0
3  1
4  1
;
```

The optimal of the lagrangian relaxation provides a lower bound of 53 on the true optimal of 45.

f.) *Repeat part (d) with* $v_1 = 200$, $v_2 = 100$.

min $15x_{1.1} + 10x_{1.2} + 30x_{2.1} + 20x_{2.2} + 200(1 - x_{1.1} - x_{1.2}) + 100(1 - x_{2.1} + x_{2.2})$
s.t. $30x_{1.1} + 50x_{2.1} \leq 80$
$30x_{1.2} + 50x_{2.2} \leq 60$
$x_{1.1}, x_{1.2}, x_{2.1}, x_{2.2} = 0$ or 1

AMPL Code
AMPL Model:

```
param num_vars = 4;

param objective_values{i in 1..num_vars};

var x{i in 1..num_vars} binary;

maximize General_Assign:
sum{i in 1..num_vars}objective_values[i]*x[i] + 200*(1-x[1] - x[2]) + 100*(1-x[3] - x

subject to Constraint_Three:
30*x[1] + 50*x[3] <= 80;

subject to Constraint_Four:
30*x[2] + 50*x[4] <= 60;

option relax_integrality 0;
```

AMPL Data File:

```
param objective_values := 1 15 2 10 3 30 4 20;
```

AMPL Run File:

```
reset;
reset;
model problem4mod.mod;
data problem4dat.dat;
option solver gurobi;
solve;
```

AMPL Solution

```
ampl: include run.run;
Gurobi 5.5.0: optimal solution; objective 300
x [*] :=
1  0
2  0
3  0
4  0
;
```

The optimal of the lagrangian relaxation provides a lower bound of 300 on the true optimal of 45.